

CUSTOMER NO.: 24498
Serial No.: 10/518,226
Office Action dated: 09/29/06
Response dated: 11/14/06

PATENT
PU020288

Amendments To The Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing Of The Claims

1. (currently amended) A method of correcting errors in a data block having N rows and X columns, comprising:
performing a parity check for each one of said N rows of said data block;
performing a parity check for each one of said X columns of said data block, the parity check for each of said N rows and said X columns for detecting bit-level errors;
identifying at least one bad byte for said data block using a byte-level error detection process;

identifying, from said parity check for each one of said N rows of said data block, from said parity check for each one of said X columns of said data ~~block~~ block, and from said at least one identified bad bytes for said data block, at least one error in said data block;
and

correcting ~~each one of~~ said at least one identified error in said data ~~block~~ block,

wherein:

the parity check for each one of said N rows of said data block is performed based on a row-based parity encoding that encodes each of said N rows.

the parity check for each one of said X columns of said data block is performed based on a column-based parity encoding, separate from the row-based parity encoding, that encodes each of said X columns, and

identifying the at least one bad byte is based on a byte-based encoding that is separate from both the row-based parity encoding and the column-based parity encoding.

2. (previously presented) The method of claim 1, wherein said at least one bad byte for said data block is identified using at least one of an 8B/10B encoding process and an 8B/10B decoding process.

3. (original) The method of claim 2, wherein N equals 32.

CUSTOMER NO.: 24498
Serial No.: 10/518,226
Office Action dated: 09/29/06
Response dated: 11/14/06

PATENT
PU020288

4. (previously presented) The method of claim 1, wherein each one of said at least one identified bad byte is located in one of said N rows of said data block.

5. (previously presented) The method of claim 4, wherein identifying at least one error in said data block further comprises:

identifying at least one suspect bit for each one of said at least one bad byte of said data block, each one of said at least one identified suspect bit being located in one of said X columns of said data block; and

determining, for each one of said at least one suspect bit, if a parity bit for a corresponding one of said X columns of said data block confirms that said suspect bit is suspect.

6. (previously presented) The method of claim 5, wherein correcting each one of said at least one identified error in said data block further comprises:

for each confirmed suspect bit, determining if at least one additional bit in said corresponding one of said X columns of said data block is also suspect; and

if no other bit in said corresponding one of said X columns of said data block is also suspect, correcting said confirmed suspect bit.

7. (previously presented) The method of claim 4, and further comprising identifying at least one suspect bit for each one of said at least one bad byte of said data block, each one of said at least one identified suspect bit being located in one of said X columns of said data block.

8. (previously presented) The method of claim 7, wherein identifying at least one suspect bit for each one of said at least one bad byte for said data block further comprises:

performing a parity check for each row of said data block in which one or more of said at least one bad byte is located; and

identifying suspect bits for each said parity checked row;

wherein said identified suspect bits for said parity checked row are said identified suspect bits for said bad byte.

CUSTOMER NO.: 24498
Serial No.: 10/518,226
Office Action dated: 09/29/06
Response dated: 11/14/06

PATENT
PU020288

9. (previously presented) The method of claim 8, wherein identifying at least one error in said data block further comprises:

identifying at least one suspect bit for each one of said at least one bad byte of said data block, each one of said at least one identified suspect bit being located in one of said X columns of said data block; and

determining, for each one of said at least one suspect bit, if a parity bit for a corresponding one of said X columns of said data block confirms that said suspect bit is suspect.

10. (previously presented) The method of claim 9, wherein correcting each one of said at least one identified error in said data block further comprises:

for each confirmed suspect bit, determining if at least one additional bit in said corresponding one of said X columns of said data block is also suspect; and

if no other bit in said corresponding one of said X columns of said data block is also suspect, correcting said confirmed suspect bit.

11. (previously presented) The method of claim 10, and further comprising:

for each said corrected suspect bit, correcting said suspect bit of said parity byte for a corresponding row of said data block.

12. (currently amended) A method of correcting errors in a data block, comprising:

(a) identifying at least one suspect bit in said data block, ~~each one of~~ said at least one suspect bit indicating a possible error in said data block;

(b) classifying ~~each one of~~ said at least one suspect bit as either a confirmed error bit or as an unconfirmed error bit; and

(c) correcting ~~each one of~~ said at least one suspect bit if said at least one suspect bit is classified as a confirmed error bit, wherein wherein:

said identifying step identifies the said at least one suspect bit using information derived from a byte-level error detection process and information derived from a bit-level error detection process, ~~and~~

~~wherein said confirmed error bits are identified using a combination of information derived from 8B/10B encoding of said data block, information derived from parity~~

CUSTOMER NO.: 24498
Serial No.: 10/518,226
Office Action dated: 09/29/06
Response dated: 11/14/06

PATENT
PU020288

~~wherein said confirmed error bits are identified using a combination of information derived from 8B/10B encoding of said data block, information derived from parity encoding along each row of said data block and information derived from parity encoding along each column of said data block~~

said byte-level error detection process is based on a byte-based encoding of individual bytes, and

said bit-level error detection process is based on a parity encoding, separate from the byte-based encoding.

13. (previously presented) The method of claim 12, and further comprising:
(d) repeating (a) through (c) until all errors in said data block have been corrected.

14. Cancelled.

15. (previously presented) The method of claim 12, wherein identifying at least one suspect bit of said data block further comprises identifying at least one bad byte of said data block using information derived from said 8B/10B encoding of said data block.

16. (previously presented) The method of claim 15, wherein classifying each one of said at least one suspect bit as either a confirmed error bit or as an unconfirmed error bit further comprises:

identifying, using information derived from parity encoding of a row of said data block in which a first one of said at least one bad byte is located, at least one suspect bit for said first one of said at least one bad byte.

17. (previously presented) The method of claim 16, wherein a first one of said at least one suspect bit for said first one of said at least one bad byte is classified as either a confirmed error bit or as an unconfirmed error bit using information derived from parity encoding of a column of said data block in said first suspect bit is located.

CUSTOMER NO.: 24498
Serial No.: 10/518,226
Office Action dated: 09/29/06
Response dated: 11/14/06

PATENT
PU020288

18. (previously presented) The method of claim 12, and further comprising the steps of:

applying a pre-selected number of iterations of an error-correction routine to said data block;

said error-correction routine being comprised of (a), (b) and (c).

19. (original) The method of claim 18, wherein said pre-selected number is two.

20. (previously presented) The method of claim 19, wherein said confirmed error bits are identified using a combination of information derived from 8B/10B encoding of said data block, information derived from parity encoding along each row of said data block and information derived from parity encoding along each column of said data block.

21. (previously presented) The method of claim 20, wherein identifying at least one suspect bit of said data block further comprises identifying at least one bad byte of said data block using information derived from said 8B/10B encoding of said data block.

22. (previously presented) The method of claim 21, wherein classifying each one of said at least one suspect bit as either a confirmed error bit or as an unconfirmed error bit further comprises:

identifying, using information derived from parity encoding of a row of said data block in which a first one of said at least one bad byte is located, at least one suspect bit for said first one of said at least one bad byte.

23. (previously presented) The method of claim 22, wherein a first one of said at least one suspect bit for said first one of said at least one bad byte is classified as either a confirmed error bit or as an unconfirmed error bit using information derived from parity encoding of a column of said data block in said first suspect bit is located.

CUSTOMER NO.: 24498
Serial No.: 10/518,226
Office Action dated: 09/29/06
Response dated: 11/14/06

PATENT
PU020288

24. (new) The method of claim 1, wherein:
the row-based parity encoding and the column-based parity encoding are performed prior to the byte-based encoding, and

the byte-based encoding encodes both the data block and parity bits from the row-based parity encoding and from the column-based parity encoding.

25. (new) The method of claim 1, wherein identifying said at least one error comprises:

selecting a bit that is in said at least one bad byte;

determining that the selected bit is indicated as being a suspect bit by at least one of the two parity checks; and

identifying the selected bit as said at least one error based on the determination that the bit is indicated as a suspect bit.

26. (new) The method of claim 25, wherein identifying said bit that is in said at least one bad byte and that is indicated as being a suspect bit by at least one of the two parity checks comprises identifying a bit that is indicated as being a suspect bit by both of the two parity checks.

27. (new) The method of claim 12, wherein:

the parity encoding is performed prior to the byte-based encoding, and

the byte-based encoding encodes both the data block and parity bits from the parity encoding.

28. (new) The method of claim 12, wherein the parity encoding comprises parity encoding each row of said data block and parity encoding each column of said data block.

CUSTOMER NO.: 24498
Serial No.: 10/518,226
Office Action dated: 09/29/06
Response dated: 11/14/06

PATENT
PU020288

29. (new) The method of claim 12, wherein identifying said at least one suspect bit comprises:

determining that a particular bit is identified as a potential error by both the byte-level error detection process and the bit-level error detection process; and

identifying the particular bit as said at least one suspect bit based on the determination that the particular bit is identified as a potential error by both the byte-level error detection process and the bit-level error detection process.

30. (new) The method of claim 12, wherein identifying said at least one suspect bit in said data block comprises:

(d) identifying all suspect bits in said data block, and

the method further comprises:

performing (b) and (c) for all identified suspect bits in said data block; and

iterating by performing (d), (b), and (c) repeatedly until a stopping condition is satisfied.

31. (new) The method of claim 30, wherein the stopping condition comprises one or more of (1) iterating a preselected number of times, (2) iterating until there are no more suspect bits, and (3) iterating until a preselected percentage of errors are corrected.